

# ROS 프로그래밍 활용



혁신성장 청년인재  
집중양성사업단



# 『프로젝트 지향 자율주행차 전문인력 양성과정』

## 1. 교육개요

- 로봇 소프트웨어 플랫폼인 ROS(Robot Operating System)을 소개하고, ROS의 기본 구성과 필요성을 설명한다.
- ROS를 설치하고, ROS의 메시지 통신 중 하나인 Topic을 실습해본다.
- 가장 기본적인 시뮬레이터 예제인 Turtle과 Rviz와 RQT 등의 ROS 기반 시각화 툴을 소개하고 실습해본다.
- 시뮬레이터와 간단한 Topic 통신을 실습해본다.

## 2. 학습목표

- ROS/ROS bridge 설치 방법 습득
- ROS의 메시지 통신 이해 및 실습
- Turtlesim Tutorial로 ROS의 기본 사용법 습득
- RC CAR 시뮬레이터와 ROS 간 메시지 통신 및 센서데이터 수신 실습
- Rviz와 RQT를 사용한 시각화 실습

## 3. 수업내용

- 5/18 ROS의 이해 및 환경 구성
- 5/19 ROS 패키지 생성 및 메시지 통신
- 5/20 ROS Turtlesim Tutorial
- 5/21 Rviz/RQT 활용
- 5/22 ROS-RCcar 시뮬레이터 통신



혁신성장 청년인재 집중양성 사업단



# ROS tutorial

프로젝트 지향 자율주행차 전문인력 양성과정

## 목차

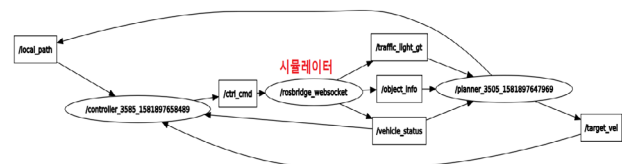
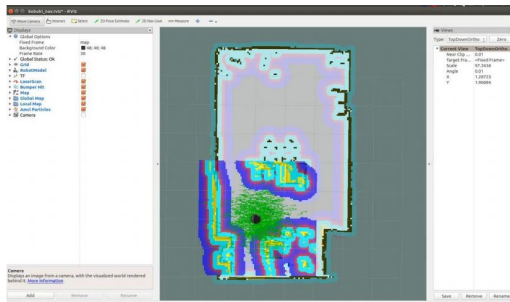
---

1. ROS
  - ROS 란
  - ROS의 기본 틀
2. ROS 설치 및 환경 세팅
  - ROS melodic 설치
  - vscode 설치

## 1. ROS

### ROS(Robot Operating System)

- ROS란
  - 로봇 소프트웨어를 개발하기 위한 소프트웨어 Framework
  - 노드간 메시지 교환 방법으로 복잡한 프로그램을 나눠 공동 개발 용이
  - 메시지 기록, 재생 기능으로 반복적인 실험 가능, 알고리즘 개발에 용이
  - 강력한 시각화 도구
  - c++, python 지원
  - 로봇 관련 다양한 기능 제공(센서 드라이버, 표준 메시지, Navigation Stack)



<https://robotbef.tistory.com/102>

# ROS(Robot Operating System)

- ROS란
  - Ford, LG, 만도, 네이버랩스 등에서 사용.



## Ford AV Dataset Tutorial

This Tutorial contains installation instructions for the packages released along with Ford Multi AV Dataset. For more details please visit the website.

### Website

To get more details about the Ford AV Dataset, please visit [avdata.ford.com](http://avdata.ford.com)

### System Requirements

This code repository has been tested on a Laptop containing 32 gb RAM, Ubuntu 16.04 and ROS Kinetic.

### Installation

#### Dependencies

These packages depend on

- python 2.x
- Standard ROS packages (roscpp, rospy, sensor\_msgs, std\_msgs, tf2\_ros)
- rviz
- pcl (>=1.7)
- pcl\_conversions
- velodyne

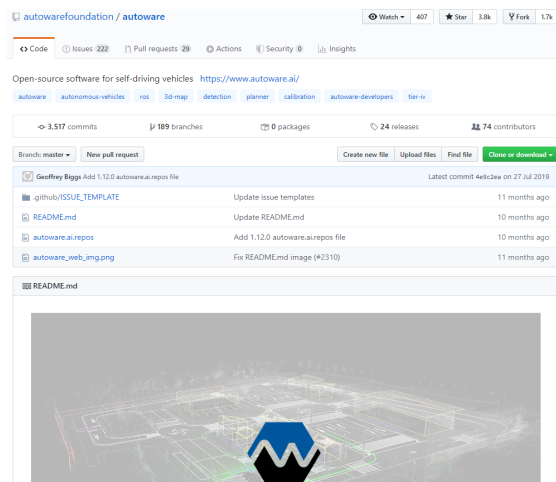
Clone the latest version from this repository into your catkin workspace and compile the packages using the following snippet. If you do not have a catkin workspace, please read [this tutorial](#) to create one.

```
cd catkin_ws/src
git clone https://github.com/Ford/AVData.git
cd ../
catkin_make
source devel/setup.bash
```

<https://github.com/Ford/AVData>

# ROS(Robot Operating System)

- ROS란
  - Autware : ros 기반 자율주행용 소프트웨어



<https://gitlab.com/autwarefoundation/autware.ai/autware/-/wikis/home>

## 2. ROS 설치 및 환경 세팅

### ROS 설치 및 환경 세팅

---

- ROS 설치 (ros-melodic 18.04 기준)
  - 터미널창을 열고 다음 명령어를 복사 붙여넣기

```
$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
$ sudo apt update
$ sudo apt install ros-melodic-desktop-full
$ sudo rosdep init
$ rosdep update
$ echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
$ source ~/.bashrc
$ source /opt/ros/melodic/setup.bash
$ sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential ros-melodic-rosbridge-suite
http://wiki.ros.org/melodic/Installation/Ubuntu
```



## ROS 설치 및 환경 세팅

- ROS 설치 (ros-melodic 18.04 기준)

- 프로젝트 세팅

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
```

gedit 편집기를 열어 bashrc를 다음과 같이 수정  
\$ gedit ~/.bashrc

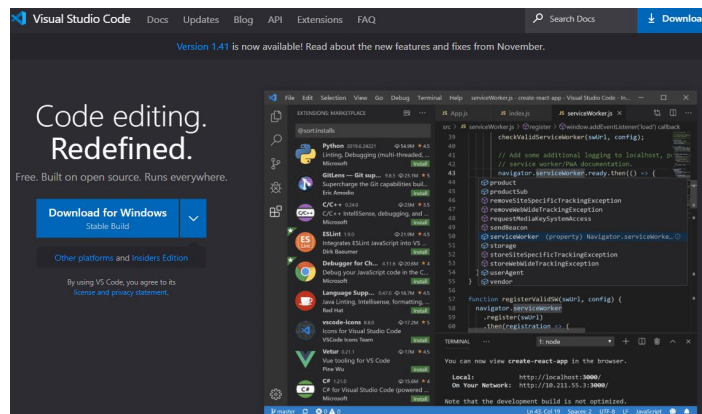
```
source ~/catkin_ws/devel/setup.bash
alias cm='cd ~/catkin_ws && catkin_make'
```

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq postcmd; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
source /opt/ros/melodic/setup.bash
source ~/catkin_ws/devel/setup.bash
alias cm='cd ~/catkin_ws && catkin_make'
```

## ROS 설치 및 환경 세팅

- VS code

- 1) Ubuntu/window에서도 쓸 수 있는 editor.
- 2) 무료이며 visual studio 보다 단순하고, C, C#, Python, Ruby과 같이 다양한 언어에 대해 빌드 가능.



## ROS 설치 및 환경 세팅

---

- VS code 설치

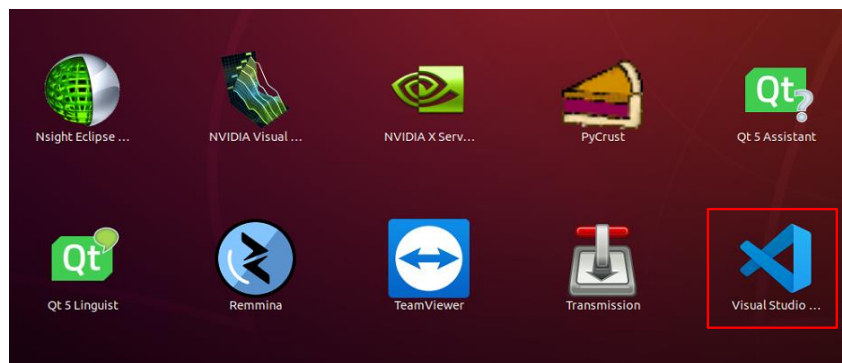
- 1) 터미널 열기.
- 2) `sudo apt-get install curl` (curl 설치)
- 3) `sudo sh -c 'curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > /etc/apt/trusted.gpg.d/microsoft.gpg'` (마이크로소프트 GPG 키를 다운로드 하여 /etc/apt/trusted.gpg.d/ 경로에 복사)
- 4) `sudo sh -c 'echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main" > /etc/apt/sources.list.d/vscode.list'` (vscode를 받기위한 저장소 추가)
- 5) `sudo apt-get update` (저장소에 패키지 목록 가져오기)
- 6) `sudo apt-get install code` (설치)
- 7) `sudo rm /etc/apt/sources.list.d/vscode.list` (설치가 끝난 후 저장소 삭제)

## ROS 설치 및 환경 세팅

---

- VS code 설치

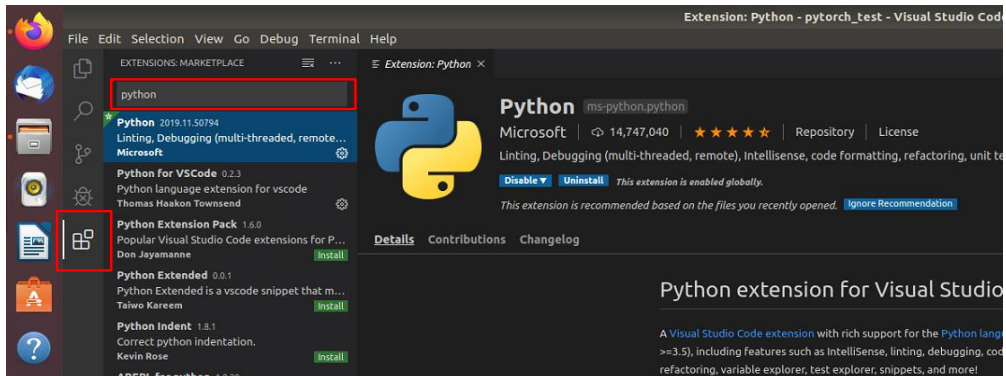
- 8) 정상적으로 설치가 되었으면 다음과 같이 vscode를 확인할 수 있다.



## ROS 설치 및 환경 세팅

- VS code 설치

9) VS code 를 실행하여, extension 창에 들어가서 python 검색 후 install 클릭



END

# ROS tutorial 2

MORAI

## 목차

---

1. ROS 통신 기본 개념
  - ROS 기본개념
  - Message
  - Topic
  - Subscriber
  - Publisher
2. Simple ROS Topic example 1
  - Pkg 생성
  - Publisher node
  - Subscriber node
  - 실행
3. Simple ROS Topic example 2 : custom message
  - Pkg 생성
  - Custom msg 파일 작성
4. Simple ROS Topic example 3 : PID control
  - plant node
  - controller node

## 1. ROS 통신 기본 개념

### ROS 통신 기본 개념

---

- Rosmaster
  - 노드와 노드 사이의 연결과 통신을 위한 서버
- node
  - ROS 에서 최소 단위의 실행 프로세서
- message
  - 노드와 노드 간의 데이터를 주고받는 양식.
- package
  - ROS 소프트웨어의 기본 단위. 패키지는 노드, 라이브러리, 환경설정 파일들을 통합하는 최소의 빌드 단위이며, 배포 단위. Ex) turtlesim

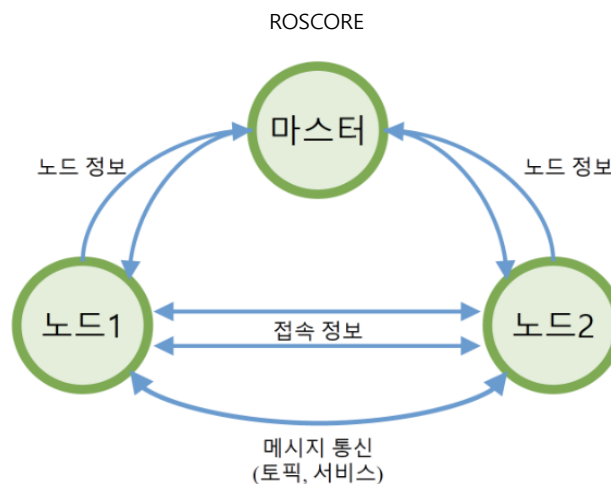
## ROS 통신 기본 개념

---

- topic
  - 단방향의 연속적인 메시지 송수신 방식. 송/수신 하고자 하는 토픽을 마스터에 등록 후 송 수신
- service
  - 양방향의 일회성 메시지 송수신 방식.
- publish/publisher
  - publish : topic에 원하는 메시지를 담아 송신하는 것.
  - publisher : publish를 수행, 하나의 노드에 여러 개의 publish가 수행 될 수 있음
- subscribe/subscriber
  - subscribe : 원하는 메시지를 담은 topic을 수신하는 것
  - subscriber : subscribe를 수행, 하나의 노드에 여러 개의 subscribe가 수행 될 수 있음

## ROS 통신 기본 개념

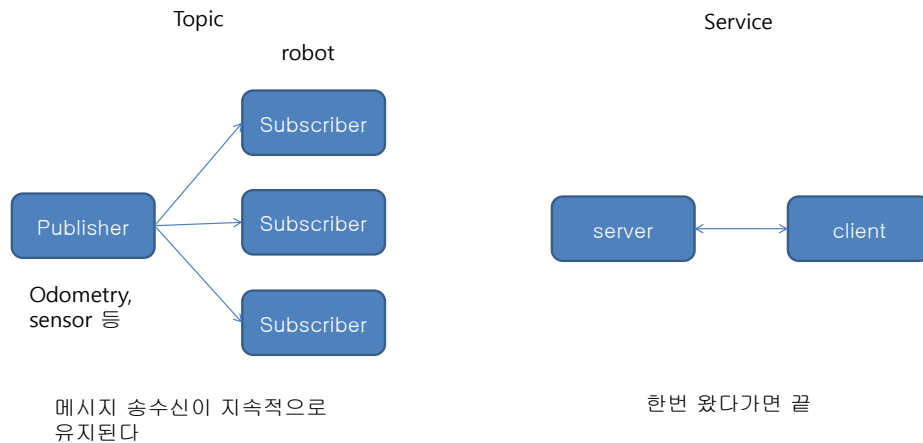
---



<https://robertchoi.gitbook.io/ros/build>

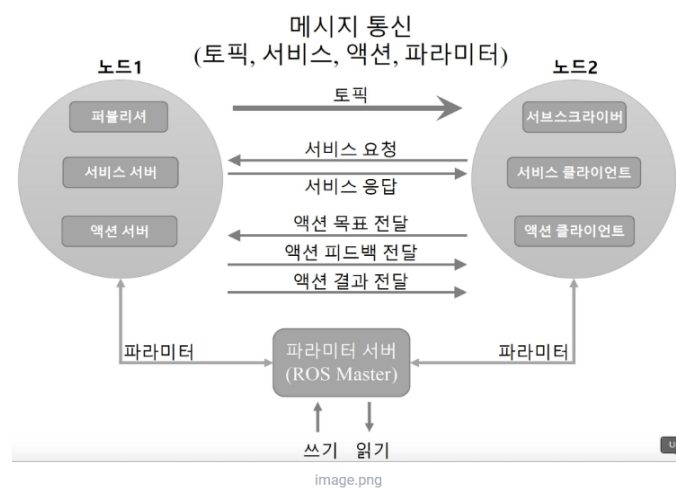
## ROS 통신 기본 개념

- Topic/Service 비교



## ROS 통신 기본 개념

- Topic/Service 비교

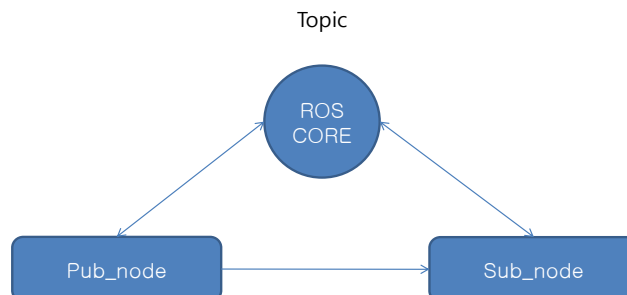


## 2. Simple ROS Topic example

### Simple ROS Topic example 1

---

- Topic example
  - 만들고자 하는 topic은 pub1, sub1 두 노드로 구성된 간단한 구조.
  - Pub은 string을 송신.
  - Sub는 pub이 송신하는 string 메시지를 받아서 console 창에 print.





## Simple ROS Topic example 1

- 기본 Message

- 일반적으로 기본적인 string 이나 int, float, bool은 std\_msgs에 있다.

### 2. ROS Message Types

ROS Message Types	
Bool	Int8
Byte	Int8MultiArray
ByteMultiArray	MultiArrayDimension
Char	MultiArrayLayout
ColorRGBA	String
Duration	Time
Empty	UInt16
Float32	UInt16MultiArray
Float32MultiArray	UInt32
Float64	UInt32MultiArray
Float64MultiArray	UInt64
Header	UInt64MultiArray
Int16	UInt8
Int16MultiArray	UInt8MultiArray
Int32	
Int32MultiArray	
Int64	
Int64MultiArray	

[http://wiki.ros.org/std\\_msgs](http://wiki.ros.org/std_msgs)

- 간단하게 string만 써보자.

### std\_msgs/String Message

File: `std_msgs/String.msg`

Raw Message Definition

string data

Compact Message Definition

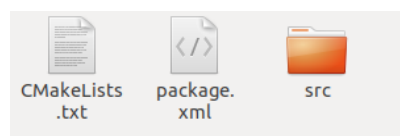
string data

autogenerated on Thu, 13 Feb 2020 04:02:12

## Simple ROS Topic example 1

- Package create

- `$ cd catkin_ws/src`
- `$ catkin_create_pkg topic_tutorial rospy std_msgs`
- 아래와 같이 src 폴더와 cmakeLists.txt, package.xml이 생성된다.



- `$ cd catkin_ws`
- `$ catkin_make`

## Simple ROS Topic example 1

---

- Pub\_node.py
  - 아래와 같이 src/pub\_node.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x pub_node.py`

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

NAME_TOPIC = '/msgs_talk'
NAME_NODE = 'pub_node'

if __name__ == '__main__':
    pub = rospy.Publisher(NAME_TOPIC, String, queue_size=10)
    rospy.init_node(NAME_NODE, anonymous=True)

    rate = rospy.Rate(10) # 10hz
    msgs_pub = String()
    while not rospy.is_shutdown():
        msgs_pub.data = "hello ROS world"
        pub.publish(msgs_pub)
        rate.sleep()
```

## Simple ROS Topic example 1

---

- Sub\_node.py
  - 아래와 같이 src/sub\_node.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x sub_node.py`

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String

NAME_TOPIC = '/msgs_talk'
NAME_NODE = 'pub_node'

def callback(msgs):
    rospy.loginfo(msgs.data)

if __name__ == '__main__':
    rospy.init_node(NAME_NODE, anonymous=True)

    sub = rospy.Subscriber(NAME_TOPIC, String, callback)
    rospy.spin()
```

## Simple ROS Topic example 1

- Topic 실행.
  - 터미널을 열고 roscore 실행

```
roscore http://10.10.120.215:11311/
File Edit View Search Terminal Help
sjyoon@sjyoon-VirtualBox:~$ roscore
... logging to /home/sjyoon/.ros/log/0339cabc-9362-11ea-8614-08002743d974/roslaunch-
sjyoon-VirtualBox-22511.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://10.10.120.215:46565/
ros_comm version 1.14.3

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.3

NODES
auto-starting new master
process[master]: started with pid [22521]
ROS_MASTER_URI=http://10.10.120.215:11311/

setting /run_id to 0339cabc-9362-11ea-8614-08002743d974
process[roscout-1]: started with pid [22533]
started core service [/roscout]
```

## Simple ROS Topic example 1

- Topic 실행.
  - topic\_tutorial/scripts 의 pub\_node.py 실행

```
sjyoon@sjyoon-VirtualBox: ~/catkin_ws/src/topic_tutorial/src
File Edit View Search Terminal Tabs Help
sjyoon@sjyoon-VirtualBox: ~/catkin_ws/src... x sjyoon@sjyoon-VirtualBox: ~/catkin_ws/src... x
sjyoon@sjyoon-VirtualBox:~/catkin_ws/src/topic_tutorial/src$ python pub_node.py
```

- topic\_tutorial/scripts 의 sub\_node.py 실행

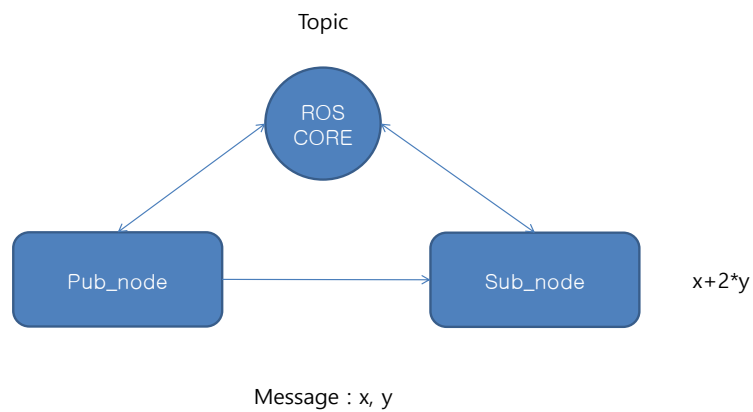
```
sjyoon@sjyoon-VirtualBox:~/catkin_ws/src/topic_tutorial/src$ python sub_node.py
[INFO] [1589186137.895761]: hello ROS world
[INFO] [1589186137.998006]: hello ROS world
[INFO] [1589186138.095700]: hello ROS world
[INFO] [1589186138.198470]: hello ROS world
[INFO] [1589186138.297412]: hello ROS world
[INFO] [1589186138.397004]: hello ROS world
[INFO] [1589186138.495412]: hello ROS world
[INFO] [1589186138.595019]: hello ROS world
[INFO] [1589186138.696276]: hello ROS world
[INFO] [1589186138.795237]: hello ROS world
[INFO] [1589186138.895770]: hello ROS world
[INFO] [1589186138.995244]: hello ROS world
[INFO] [1589186139.096997]: hello ROS world
```

### 3. Simple ROS Topic example 2

#### Simple ROS Topic example 2

---

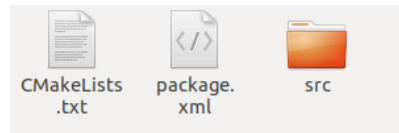
- Topic example
  - 메시지를 직접 커스터마이징 해보자.
  - 메시지 안은 x,y 라는 두 float64 변수로 구성되어 있다.
  - Sub node 에서 메시지를 받아서  $x+2*y$ 를 프린트하자.



## Simple ROS Topic example 2

---

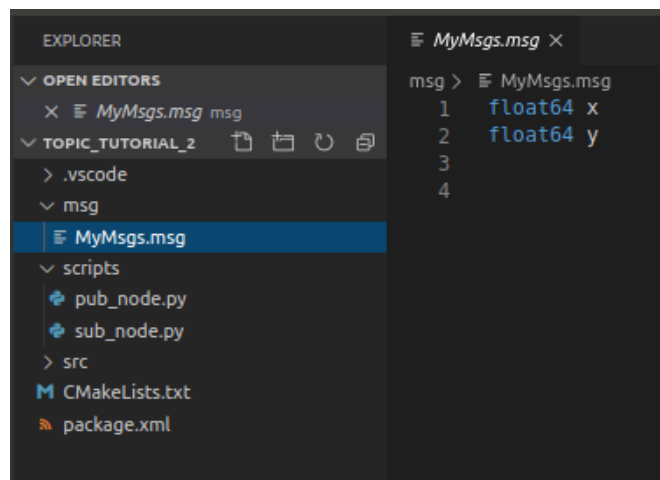
- Package create
  - \$ cd catkin\_ws/src
  - \$ catkin\_create\_pkg topic\_tutorial\_2 rospy std\_msgs message\_generation message\_runtime
  - 아래와 같이 src 폴더와 cmakeLists.txt, package.xml이 생성된다.



## Simple ROS Topic example 2

---

- Package create
  - 빌드를 하기 전에 메시지를 만든다.
  - Msg 폴더를 만들어 아래와 같이 msg 파일을 작성.



## Simple ROS Topic example 2

- Package create
  - package.xml 내부 내용이 아래와 같은지 확인하고, 다르다면 수정.

```
49 <!-- Use doc_depend for packages you need only for building docu
50 <!-- <doc_depend>doxygen</doc_depend> -->
51 <buildtool_depend>catkin</buildtool_depend>
52 <build_depend>message_generation</build_depend>
53 <build_depend>rospy</build_depend>
54 <build_depend>std_msgs</build_depend>
55 <build_export_depend>rospy</build_export_depend>
56 <build_export_depend>std_msgs</build_export_depend>
57 <exec_depend>message_runtime</exec_depend>
58 <exec_depend>rospy</exec_depend>
59 <exec_depend>std_msgs</exec_depend>
60
61
```

## Simple ROS Topic example 2

- Package create
  - cmakeLists.txt 내부 내용이 아래와 같은지 확인하고, 다르다면 수정.

```
9 ## is used, also find other catkin packages
0 find_package(catkin REQUIRED COMPONENTS
1   message_generation
2   rospy
3   std_msgs
4 )
5
```

```
49 ## Generate messages in the 'msg' folder
50 add_message_files(
51   FILES
52   MyMsgs.msg
53 )
54
```

```
69 ## Generate added messages and services with any dep
70 generate_messages(
71   DEPENDENCIES
72   std_msgs
73   topic_tutorial_2
74 )
75
```

```
4 ## DEPENDS: system dependencies of this project that dep
5 catkin_package(
6   # INCLUDE_DIRS include
7   # LIBRARIES topic_tutorial_2
8   # CATKIN_DEPENDS message_generation rospy std_msgs
9   CATKIN_DEPENDS message_runtime
10  # DEPENDS system_lib
11 )
12
```

- \$ cd catkin\_ws
- \$ catkin\_make

## Simple ROS Topic example 2

---

- Pub\_node.py
  - 아래와 같이 scripts/pub\_node.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x pub_node.py`

```
#!/usr/bin/env python

import rospy
from topic_tutorial_2.msg import MyMsgs

NAME_TOPIC = '/msgs_talk'
NAME_NODE = 'pub_node'

if __name__ == '__main__':
    rospy.init_node(NAME_NODE, anonymous=True)

    pub = rospy.Publisher(NAME_TOPIC, MyMsgs, queue_size=10)

    rate = rospy.Rate(10) # 10hz

    msgs_pub = MyMsgs()

    while not rospy.is_shutdown():
        msgs_pub.x = 10
        msgs_pub.y = 10

        pub.publish(msgs_pub)

        rate.sleep()
```

## Simple ROS Topic example 2

---

- Sub\_node.py
  - 아래와 같이 scripts/sub\_node.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x sub_node.py`

```
#!/usr/bin/env python

import rospy
from topic_tutorial_2.msg import MyMsgs

NAME_TOPIC = '/msgs_talk'
NAME_NODE = 'pub_node'

def callback(msgs):
    rospy.loginfo(msgs.x + 2*msgs.y)

if __name__ == '__main__':
    rospy.init_node(NAME_NODE, anonymous=True)

    sub = rospy.Subscriber(NAME_TOPIC, MyMsgs, callback)

    rospy.spin()
```

## Simple ROS Topic example 2

- Topic 실행.
  - 터미널을 열고 roscore 실행

```
roscore http://10.10.120.215:11311/
File Edit View Search Terminal Help
sjoyoon@sjoyoon-VirtualBox:~$ roscore
... logging to /home/sjoyoon/.ros/log/0339cabc-9362-11ea-8614-08002743d974/roslaunch-
sjoyoon-VirtualBox-22511.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://10.10.120.215:46565/
ros_comm version 1.14.3

SUMMARY
=====
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.3
NODES
auto-starting new master
process[master]: started with pid [22521]
ROS_MASTER_URI=http://10.10.120.215:11311/

setting /run_id to 0339cabc-9362-11ea-8614-08002743d974
process[roscout-1]: started with pid [22533]
started core service [/roscout]
```

## Simple ROS Topic example 2

- Topic 실행.
  - 이번에 rosrn [패키지 이름] [노드 이름] 으로 실행해보자.
  - \$rosrn topic\_tutorial\_2 pub\_node.py
  - \$rosrn topic\_tutorial\_2 sub\_node.py 실행

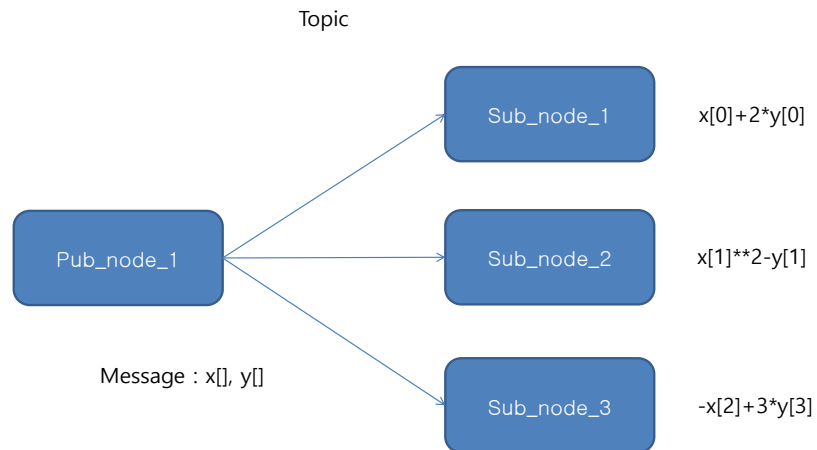
```
^Csjoyoon@sjoyoon-VirtualBox:~$ rosrn topic_tutorial_2 sub_node.py
[INFO] [1589195505.173936]: 30.0
[INFO] [1589195505.273442]: 30.0
[INFO] [1589195505.372599]: 30.0
[INFO] [1589195505.472763]: 30.0
[INFO] [1589195505.574542]: 30.0
[INFO] [1589195505.673149]: 30.0
[INFO] [1589195505.773579]: 30.0
[INFO] [1589195505.872471]: 30.0
[INFO] [1589195505.972722]: 30.0
[INFO] [1589195506.073217]: 30.0
[INFO] [1589195506.175926]: 30.0
[INFO] [1589195506.273328]: 30.0
[INFO] [1589195506.374396]: 30.0
[INFO] [1589195506.472554]: 30.0
[INFO] [1589195506.572986]: 30.0
[INFO] [1589195506.674547]: 30.0
```



## Simple ROS Topic example 2

---

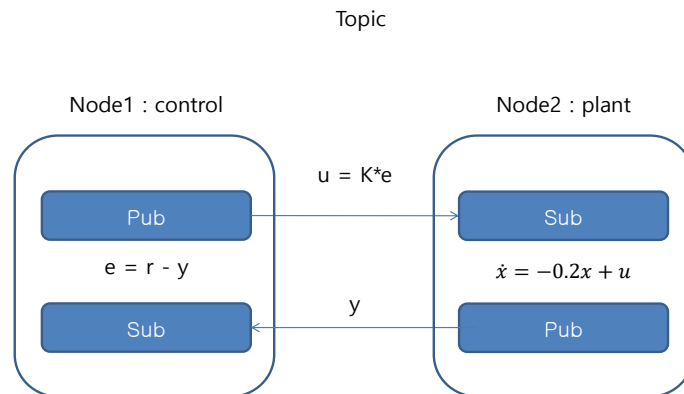
- Topic 연습문제
  - 아래와 같은 topic을 만들어 봅시다.



## 4. Simple ROS Topic example 3

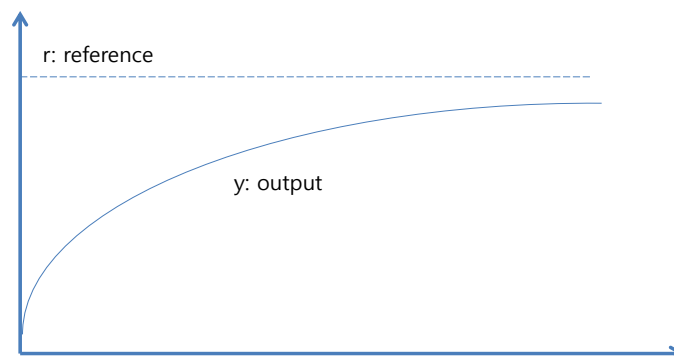
## Simple ROS Topic example 3

- Topic example : PID
  - Node 안에 pub/sub 둘이 존재.
  - Node 1은 controller.
  - Node 2은 plant.



## Simple ROS Topic example 3

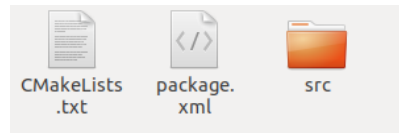
- Topic example : PID
  - 1차 plant에 대한 출력  $y$ 를  $r$ 만큼 나오도록 pid 로 컨트롤.



## Simple ROS Topic example 3

---

- Package create
  - 기존에 topic\_tutorial/ topic\_tutorial\_2 에서 진행해도 된다.



## Simple ROS Topic example 3

---

- plant.py 노드
  - 아래와 같이 scripts/plant.py 작성.
  - 작성 후 해당 디렉토리에 chmod +x plant.py

```
#!/usr/bin/env python

import rospy
from std_msgs.msg import Float32

name_node = 'plant'

class PLANT:
    def __init__(self, T=0.1, a=0.2):
        self.T = T
        self.a = a
        self.measure_msgs = Float32()
        self.ctrl_topic = '/pid_ctrl'
        self.measure_topic = '/measure_plant'

        self.x = 0.0
        self.u = 0.0

        self.sub = rospy.Subscriber(self.ctrl_topic, Float32, self.ctrl_callback)
        self.pub = rospy.Publisher(self.measure_topic, Float32, queue_size=10)

    def ctrl_callback(self, msg):
        self.u = msg.data

    def pub_measure_msgs(self):
        self.x = self.x - self.T*self.a*self.x + self.T*self.u
        self.measure_msgs.data = self.x

        self.pub.publish(self.measure_msgs)
```

## Simple ROS Topic example 3

---

- plant.py 노드
  - 아래와 같이 scripts/plant.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x plant.py`

```
if __name__ == '__main__':  
    rospy.init_node(name_node)  
    plant = PLANT()  
    rate = rospy.Rate(10) # 10hz  
    while not rospy.is_shutdown():  
        plant.pub_measure_msgs()  
        rate.sleep()
```

## Simple ROS Topic example 3

---

- controller.py 노드
  - 아래와 같이 scripts/controller.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x controller.py`

```
#!/usr/bin/env python  
import rospy  
from std_msgs.msg import Float32  
name_node = 'controller'  
  
class PController:  
    def __init__(self, ref=10, T=0.1, K=1):  
        self.T = T  
        self.ref = ref  
        self.K = K  
        self.ctrl_msgs = Float32()  
        self.ctrl_topic = '/pid_ctrl'  
        self.measure_topic = '/measure_plant'  
  
        self.y = None  
  
        self.sub = rospy.Subscriber(self.measure_topic, Float32, self.measure_callback)  
        self.pub = rospy.Publisher(self.ctrl_topic, Float32, queue_size=10)  
  
    def measure_callback(self, msgs):  
        self.y = msgs.data  
  
    def pub_ctrl_msgs(self):  
        err = self.ref - self.y  
        u = self.K * err  
        self.ctrl_msgs.data = u  
        self.pub.publish(self.ctrl_msgs)
```

## Simple ROS Topic example 3

---

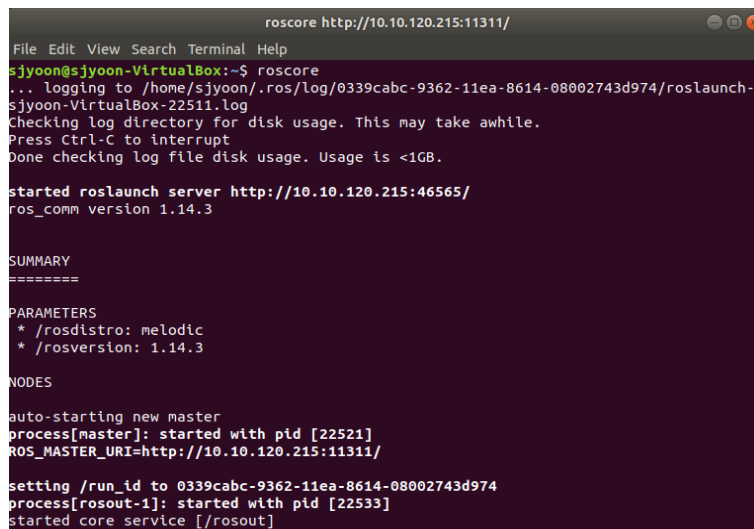
- controller.py 노드
  - 아래와 같이 scripts/controller.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x controller.py`

```
if __name__ == '__main__':  
    rospy.init_node(name_node)  
  
    plant = PLANT()  
  
    rate = rospy.Rate(10) # 10hz  
  
    while not rospy.is_shutdown():  
        plant.pub_measure_msgs()  
        rate.sleep()
```

## Simple ROS Topic example 3

---

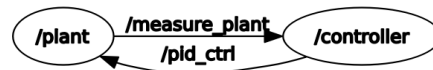
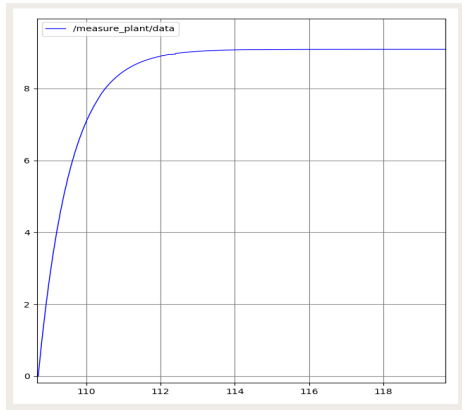
- Topic 실행.
  - 터미널을 열고 roscore 실행

A terminal window titled 'roscore http://10.10.120.215:11311/' showing the execution of the 'roscore' command. The terminal output includes logging information, disk usage checks, and the successful start of the roscore master service. It also displays summary, parameters, and nodes sections.

```
roscore http://10.10.120.215:11311/  
File Edit View Search Terminal Help  
sjyoon@sjyoon-VirtualBox:~$ roscore  
... logging to /home/sjyoon/.ros/log/0339cab9-9362-11ea-8614-08002743d974/roslaunch-  
sjyoon-VirtualBox-22511.log  
checking log directory for disk usage. This may take awhile.  
Press Ctrl-C to interrupt  
Done checking log file disk usage. Usage is <1GB.  
  
started roslaunch server http://10.10.120.215:46565/  
ros_comm version 1.14.3  
  
SUMMARY  
=====  
  
PARAMETERS  
* /roscpp: melodic  
* /rosversion: 1.14.3  
  
NODES  
  
auto-starting new master  
process[master]: started with pid [22521]  
ROS_MASTER_URI=http://10.10.120.215:11311/  
  
setting /run_id to 0339cab9-9362-11ea-8614-08002743d974  
process[roscpp-1]: started with pid [22533]  
started core service [/roscpp]
```

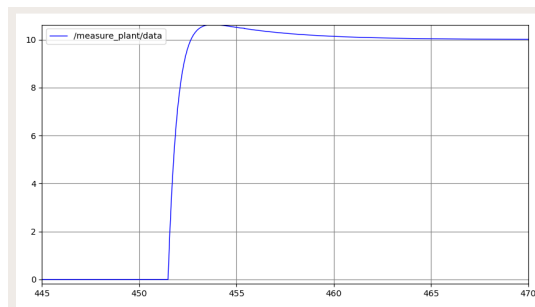
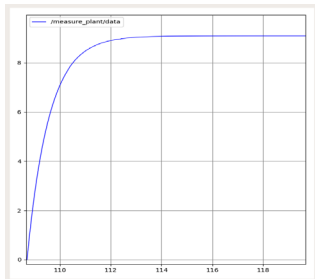
## Simple ROS Topic example 3

- Topic 실행.
  - 이번에 rosrun [패키지 이름] [노드 이름] 으로 실행해보자.
  - `$roslaunch topic_tutorial_2 plant.py`
  - `$rqt`
  - `$roslaunch topic_tutorial_2 controller.py` 실행



## Simple ROS Topic example 3

- Topic 연습문제.
  - controller.py를 수정하여  $r-y = e$ 를 0으로 만들어보자.
  - Hint : pid



END

# ROS tutorial 3

MORAI

## 목차

---

1. Turtlesim
  - Turtlesim 소개
  - Teleop\_key control
  - Message of Turtlesim
2. Turtlesim control 1 : Linear motion
  - 목적
  - Msg 구성
  - Pkg 생성
  - Control node
  - 실행
  - 연습문제
3. Turtlesim control 2 : Rotate the turtle
  - 목적
  - Control node
  - 실행
  - 연습문제
4. Turtlesim Application : position control
  - 목적



## 1. Turtlesim

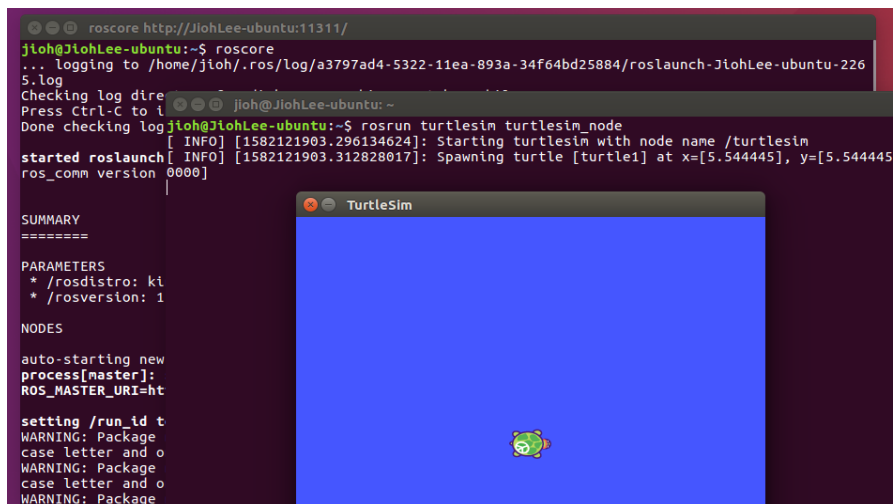
### Turtlesim

- turtlesim
  - ROS에서 tutorial에 가장 많이 사용하는 패키지
  - Link : <http://wiki.ros.org/turtlesim>

\$ roscore

\$ roslaunch turtlesim turtlesim\_node

roscore를 실행시킨 후 새로운 터미널을 열어 다음 명령어를 입력합니다. 하나의 터미널에 하나의 프로세스만 실행할 수 있습니다.



```
roscore http://jiohLee-ubuntu:11311/
jioh@jiohLee-ubuntu:~$ roscore
... logging to /home/jioh/.ros/log/a3797ad4-5322-11ea-893a-34f64bd25884/roslaunch-jiohLee-ubuntu-226
5.log
Checking log directory for log
Press Ctrl-C to interrupt
Done checking log directory logs
jioh@jiohLee-ubuntu:~$ roslaunch turtlesim turtlesim_node
[ INFO] [1582121903.296134624]: Starting turtlesim with node name /turtlesim
started roslaunch [ INFO] [1582121903.312828017]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445]
ros_comm version 0000

SUMMARY
=====

PARAMETERS
* /rostdistro: ki
* /rosversion: 1

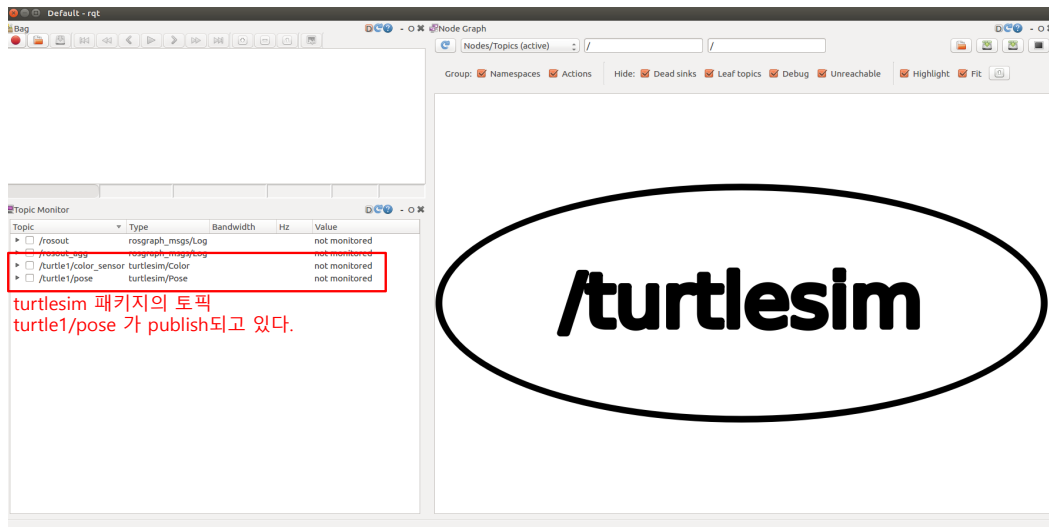
NODES

auto-starting new process[master]:
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 1582121903.312828017
WARNING: Package case letter and o
WARNING: Package case letter and o
WARNING: Package case letter and o
```

## Turtlesim

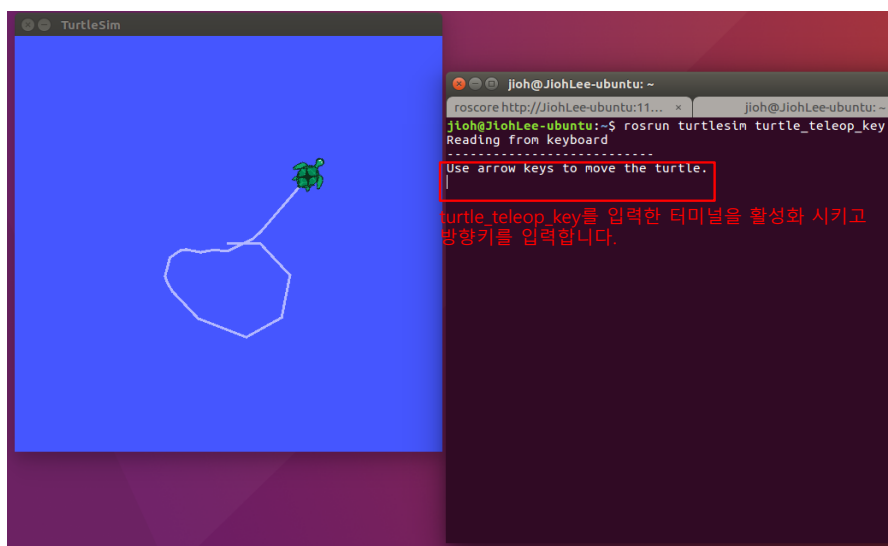
- rqt topic monitor / node graph
  - topic monitor : 현재 publish 되고있는 topic을 확인할 수 있다.
  - node graph : 현재 실행되고 있는 node와 publish/subscribe 되고 있는 토픽들의 관계를 그래프로 볼 수 있다.



## Turtlesim

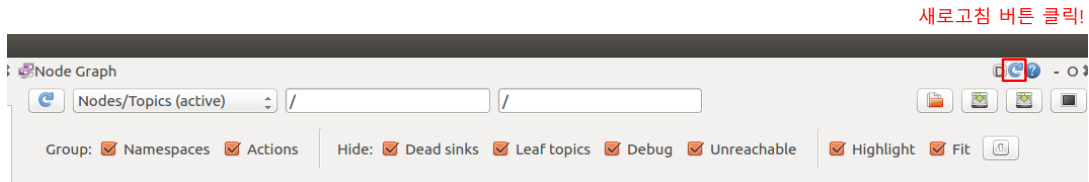
- turtle\_teleop\_key 노드 실행
  - 방향키로 turtle의 위치를 조종할 수 있도록 하는 노드

\$ rosrn turtlesim turtle\_teleop\_key



## Turtlesim

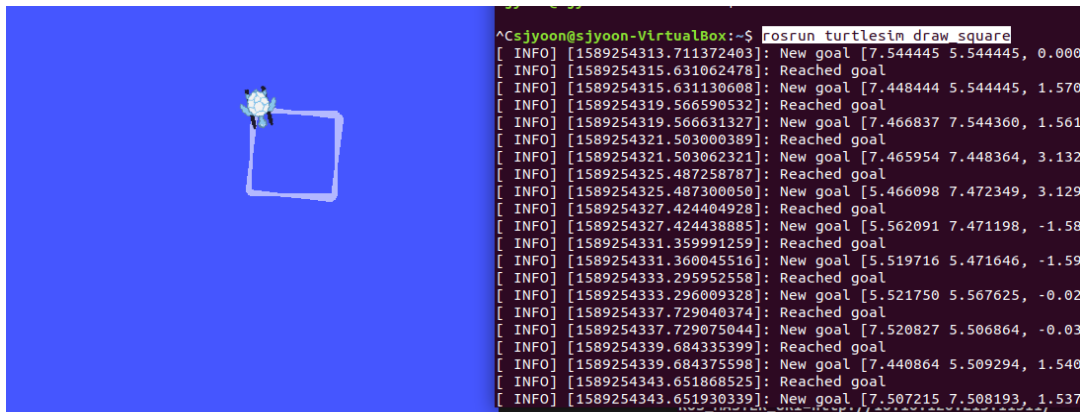
- rqt에서 node\_graph 보기



/teleop\_turtle 노드에서 토픽 /turtle1/cmd\_vel가 publish되어 /turtlesim 노드에서 subscribe하고 있다.

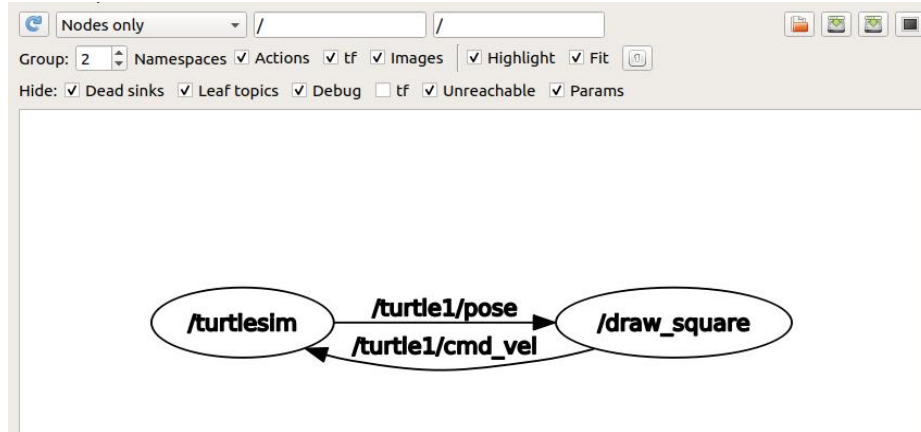
## Turtlesim

- draw\_square 노드 실행
    - Turtle이 사각형을 만들면서 이동하기 위한 노드
- \$ rosrun turtlesim draw\_square



## Turtlesim

- rqt에서 node\_graph 보기



/teleop\_turtle 노드에서 토픽 /turtle1/cmd\_vel가 publish되어 /turtlesim 노드에서 subscribe하고 있다.

## Turtlesim

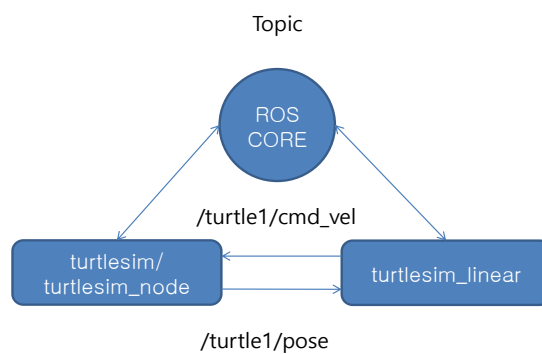
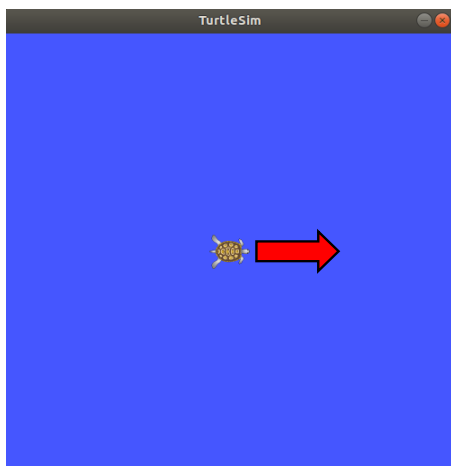
- geometry\_msgs/Point
  - 자유공간에서 위치를 나타내는 좌표, x,y,z로 구성됨
  - Link : [http://docs.ros.org/api/geometry\\_msgs/html/msg/Point.html](http://docs.ros.org/api/geometry_msgs/html/msg/Point.html)
- geometry\_msgs/Quaternion
  - 자유공간에서 방향을 사원수 형태로 나타냄
  - Link : [http://docs.ros.org/api/geometry\\_msgs/html/msg/Quaternion.html](http://docs.ros.org/api/geometry_msgs/html/msg/Quaternion.html)
- geometry\_msgs/Pose
  - Point와 Quaternion을 포함하며 자세와 위치를 한번에 나타냄
  - Link : [http://docs.ros.org/melodic/api/geometry\\_msgs/html/msg/Pose.html](http://docs.ros.org/melodic/api/geometry_msgs/html/msg/Pose.html)
- geometry\_msgs/PoseWithCovariance
  - Pose와 함께 데이터의 분산을 함께 표시함
  - Link : [http://docs.ros.org/api/geometry\\_msgs/html/msg/PoseWithCovariance.html](http://docs.ros.org/api/geometry_msgs/html/msg/PoseWithCovariance.html)
- nav\_msgs/Odometry 메시지
  - Pose와 함께 위치/방향을 벡터로 표시한 Twist를 포함, 위치와 속도를 추정하는 데이터
  - Link : [http://docs.ros.org/api/nav\\_msgs/html/msg/Odometry.html](http://docs.ros.org/api/nav_msgs/html/msg/Odometry.html)
- nav\_msgs/Path 메시지
  - 로봇이 따라갈 경로를 나타내는 위치정보의 배열
  - Link : [http://docs.ros.org/melodic/api/nav\\_msgs/html/msg/Path.html](http://docs.ros.org/melodic/api/nav_msgs/html/msg/Path.html)

## 2. Turtlesim control 1 : Linear motion

### Turtlesim control 1 : Linear motion

---

- 목적
  - Turtle을 3만큼 전진시키는 노드 작성.



## Turtlesim control 1 : Linear motion

- Msg 구성
  - turtle1/pose로 turtle의 위치와 자세 등을 받을 수 있다.
  - Geometry\_msgs/Twist 로 command를 줄수 있다.

Topic	Type	Bandwidth	Hz	Value
▶ <input type="checkbox"/> /rosout	rosgraph_msgs/Log			not monitored
▶ <input type="checkbox"/> /rosout_agg	rosgraph_msgs/Log			not monitored
▶ <input type="checkbox"/> /turtle1/color_sensor	turtlesim/Color			not monitored
▼ <input checked="" type="checkbox"/> /turtle1/pose	turtlesim/Pose	1.23KB/s	61.85	
angular_velocity	float32			0.0
linear_velocity	float32			0.0
theta	float32			0.0
x	float32			5.544444561004639
y	float32			5.544444561004639

```
sjyoon@sjyoon-VirtualBox:~$ rosmmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
float64 x
float64 y
float64 z
geometry_msgs/Vector3 angular
float64 x
float64 y
float64 z
```

## Turtlesim control 1 : Linear motion

- Package create
  - \$ cd catkin\_ws/src
  - \$ catkin\_create\_pkg turtlesim\_control rospy std\_msgs geometry\_msgs message\_generation message\_runtime
  - 아래와 같이 src 폴더와 cmakeLists.txt, package.xml이 생성된다.

```
<buildtool_depend>catkin</buildtool_depend>
<build_depend>rospy</build_depend>
<build_depend>std_msgs</build_depend>
<build_depend>geometry_msgs</build_depend>
<build_depend>message_generation</build_depend>
<build_export_depend>rospy</build_export_depend>
<build_export_depend>std_msgs</build_export_depend>
<build_export_depend>geometry_msgs</build_export_depend>
<exec_depend>message_runtime</exec_depend>
<exec_depend>rospy</exec_depend>
<exec_depend>std_msgs</exec_depend>
<exec_depend>geometry_msgs</exec_depend>
```

## Turtlesim control 1 : Linear motion

- Package create
  - \$ cd catkin\_ws/src
  - \$ catkin\_create\_pkg turtlesim\_control rospy std\_msgs geometry\_msgs message\_generation message\_runtime
  - 아래와 같이 src 폴더와 cmakeLists.txt, package.xml이 생성된다.

```
find_package(catkin REQUIRED COMPONENTS
  rospy
  std_msgs
  geometry_msgs
  message_generation
)
```

```
catkin_package(
  # INCLUDE_DIRS include
  # LIBRARIES turtlesim_control
  CATKIN_DEPENDS rospy std_msgs geometry_msgs message_runtime
  # DEPENDS system_lib
)
```

- \$ cd catkin\_ws
- \$ catkin\_make

## Turtlesim control 1 : Linear motion

- Control\_node 작성
  - 아래와 같이 scripts/turtlesim\_linear.py 작성.
  - 작성 후 해당 디렉토리에 chmod +x turtlesim\_linear.py

```
turtlesim_control > scripts > turtlesim_linear.py > {} time
1  #!/usr/bin/env python
2
3  import rospy
4  from geometry_msgs.msg import Twist
5  from turtlesim.msg import Pose
6  import math
7  import time
8
9
10 name_node = 'turtlesim_linear'
11
12 class LINEARController:
13
14     def __init__(self, ref=3, spd=1):
15         self.ref = ref
16         self.spd = spd
17         self.dist_move = 0
18         self.cmd_vel_msgs = Twist()
19         self.cmd_vel_topic = '/turtle1/cmd_vel'
20         self.pose_topic = '/turtle1/pose'
21
22         self.x = None
23         self.y = None
24         self.yaw = None
25
26         self.x0 = None
27         self.y0 = None
28         self.yaw0 = None
```

## Turtlesim control 1 : Linear motion

---

- Control\_node 작성
  - 아래와 같이 scripts/turtlesim\_linear.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x turtlesim_linear.py`

```
33
34     def pose_callback(self, msgs):
35         self.x = msgs.x
36         self.y = msgs.y
37         self.yaw = msgs.theta
38
39
40     def pub_cmd_msgs(self):
41
42         self.dist_move = math.sqrt((self.x - self.x0)**2 + (self.y - self.y0)**2)
43
44         if self.dist_move <= self.ref:
45             rospy.loginfo(str(self.dist_move))
46
47             self.cmd_vel_msgs.linear.x = abs(self.spd)
48         else:
49             self.cmd_vel_msgs.linear.x = 0
50
51         self.pub.publish(self.cmd_vel_msgs)
52
53     def init_pose(self):
54
55         self.x0 = self.x
56         self.y0 = self.y
57         self.yaw0 = self.yaw
58
```

## Turtlesim control 1 : Linear motion

---

- Control\_node 작성
  - 아래와 같이 scripts/turtlesim\_linear.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x turtlesim_linear.py`

```
59
60     if __name__ == '__main__':
61
62         rospy.init_node(name_node)
63
64         l_ctrl = LINEARController()
65
66         time.sleep(2)
67
68         rate = rospy.Rate(10) # 10hz
69
70         l_ctrl.init_pose()
71
72         while not rospy.is_shutdown():
73
74             l_ctrl.pub_cmd_msgs()
75
76             rate.sleep()
77
```



## Turtlesim control 1 : Linear motion

- Topic 실행.
  - 터미널을 열고 roscore 실행

```
roscore http://10.10.120.215:11311/
File Edit View Search Terminal Help
sjoyoon@sjoyoon-VirtualBox:~$ roscore
... logging to /home/sjoyoon/.ros/log/0339cab3-9362-11ea-8614-08002743d974/roslaunch-
sjoyoon-VirtualBox-22511.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

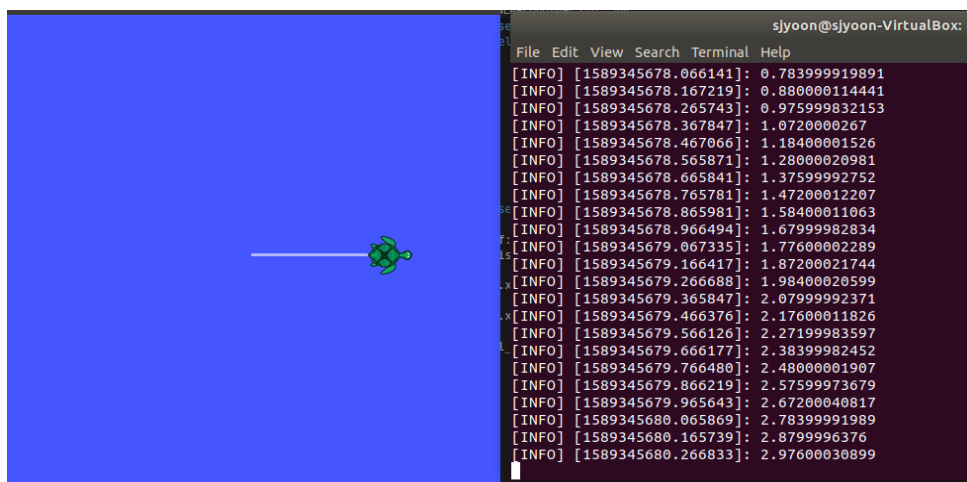
started roslaunch server http://10.10.120.215:46565/
ros_comm version 1.14.3

SUMMARY
=====
PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.3
NODES
auto-starting new master
process[master]: started with pid [22521]
ROS_MASTER_URI=http://10.10.120.215:11311/

setting /run_id to 0339cab3-9362-11ea-8614-08002743d974
process[roscout-1]: started with pid [22533]
started core service [/roscout]
```

## Turtlesim control 1 : Linear motion

- Topic 실행.
  - \$roslaunch turtlesim\_control turtlesim\_linear.py

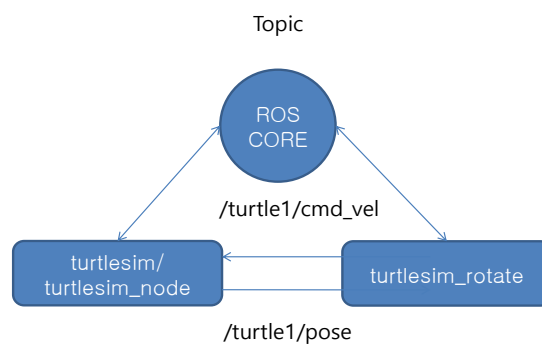
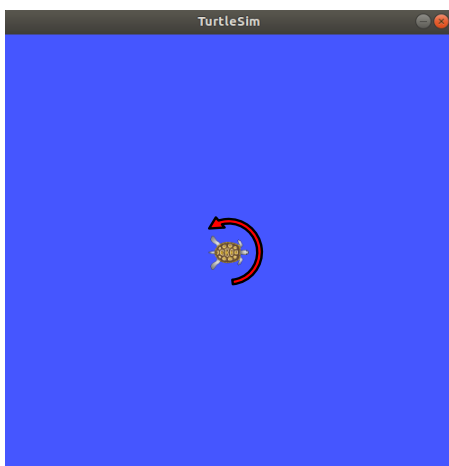


### 3. Turtlesim control 2 : Rotation

#### Turtlesim control 2 : Rotation

---

- 목적
  - Turtle을 90도 회전시키는 노드 작성.



## Turtlesim control 2 : Rotation

- Control node 작성
  - 아래와 같이 scripts/turtlesim\_rotate.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x turtlesim_rotate.py`

```
1  #!/usr/bin/env python
2
3  import rospy
4  from geometry_msgs.msg import Twist
5  from turtlesim.msg import Pose
6  import math
7  import time
8
9
10 name_node = 'turtlesim_rotate'
11
12 class ROTATEController:
13
14     def __init__(self, ref=math.pi/2, wspd=1):
15         self.ref = ref
16         self.wspd = wspd
17         self.dist_move = 0
18         self.cmd_vel_msgs = Twist()
19         self.cmd_vel_topic = '/turtle1/cmd_vel'
20         self.pose_topic = '/turtle1/pose'
21
22         self.x = None
23         self.y = None
24         self.yaw = None
25
26         self.x0 = None
27         self.y0 = None
28         self.yaw0 = None
29
30         self.sub = rospy.Subscriber(self.pose_topic, Pose, self.pose_callback)
31         self.pub = rospy.Publisher(self.cmd_vel_topic, Twist, queue_size=10)
```

## Turtlesim control 2 : Rotation

- Control node 작성
  - 아래와 같이 scripts/turtlesim\_rotate.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x turtlesim_rotate.py`

```
32
33     def pose_callback(self, msg):
34         self.x = msg.x
35         self.y = msg.y
36         self.yaw = msg.theta
37
38
39     def pub_cmd_msgs(self):
40
41         self.dist_move = self.yaw - self.yaw0
42
43         if self.dist_move <= self.ref:
44             rospy.loginfo(str(self.dist_move))
45
46             self.cmd_vel_msgs.angular.z = abs(self.wspd)
47         else:
48             self.cmd_vel_msgs.angular.z = 0
49
50         self.pub.publish(self.cmd_vel_msgs)
51
52     def init_pose(self):
53
54         self.x0 = self.x
55         self.y0 = self.y
56         self.yaw0 = self.yaw
57
```

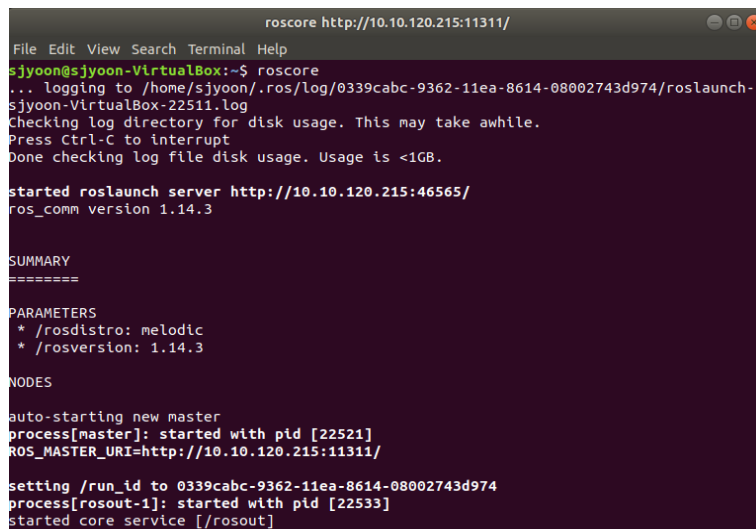
## Turtlesim control 2 : Rotation

- Control node 작성
  - 아래와 같이 scripts/turtlesim\_rotate.py 작성.
  - 작성 후 해당 디렉토리에 `chmod +x turtlesim_rotate.py`

```
59 if __name__ == '__main__':
60
61     rospy.init_node('name_node')
62
63     r_ctrl = ROTATEController()
64
65     time.sleep(2)
66
67     rate = rospy.Rate(10) # 10hz
68
69     r_ctrl.init_pose()
70
71     while not rospy.is_shutdown():
72
73         r_ctrl.pub_cmd_msgs()
74         rate.sleep()
75
76
```

## Turtlesim control 2 : Rotation

- Topic 실행.
  - 터미널을 열고 roscore 실행

A terminal window titled 'roscore http://10.10.120.215:11311/' showing the execution of the 'roscore' command. The output includes logging information, disk usage checks, and the successful start of the ROS master and core services.

```
roscore http://10.10.120.215:11311/
File Edit View Search Terminal Help
sjyoon@sjyoon-VirtualBox:~$ roscore
... logging to /home/sjyoon/.ros/log/0339cab3-9362-11ea-8614-08002743d974/roslaunch-
sjyoon-VirtualBox-22511.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://10.10.120.215:46565/
ros_comm version 1.14.3

SUMMARY
=====

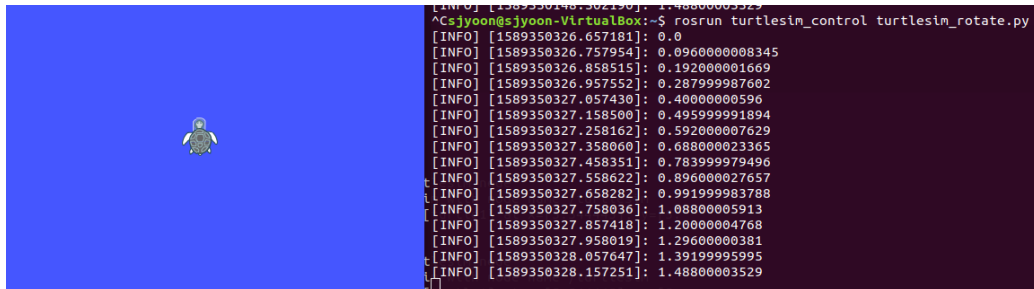
PARAMETERS
 * /roscpp: melodic
 * /rosversion: 1.14.3

NODES
auto-starting new master
process[master]: started with pid [22521]
ROS_MASTER_URI=http://10.10.120.215:11311/

setting /run_id to 0339cab3-9362-11ea-8614-08002743d974
process[roscout-1]: started with pid [22533]
started core service [/roscout]
```

## Turtlesim control 2 : Rotation

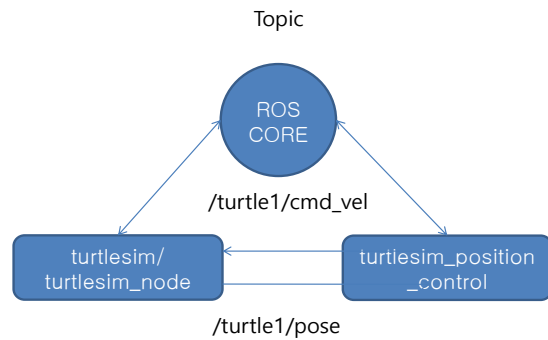
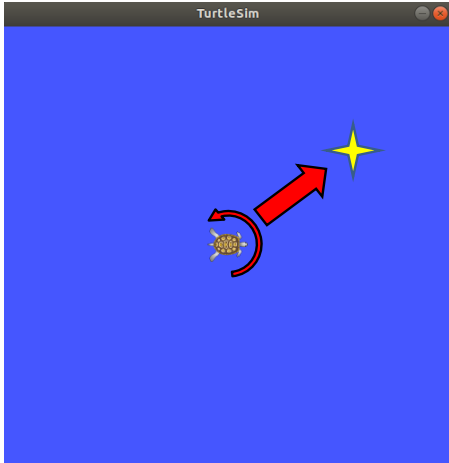
- Topic 실행.
  - `$roslaunch turtlesim_control turtlesim_rotate.py`



## 4. Turtlesim Application : position control

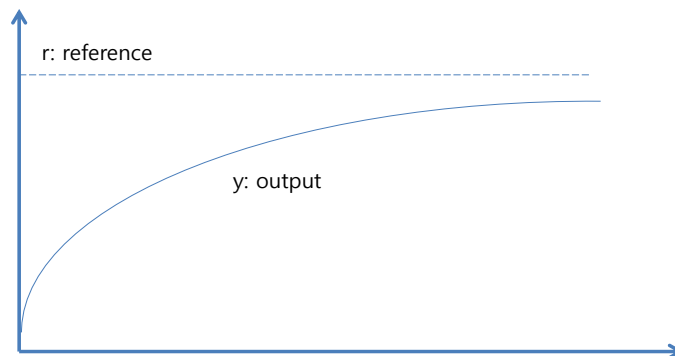
## Turtlesim Application : position control

- Turtle Sim Position
  - Target point (x,y) 좌표를 주고 이동시키기.



## Turtlesim Application : position control

- Turtle Sim Position
  - PID를 응용해서 작성.

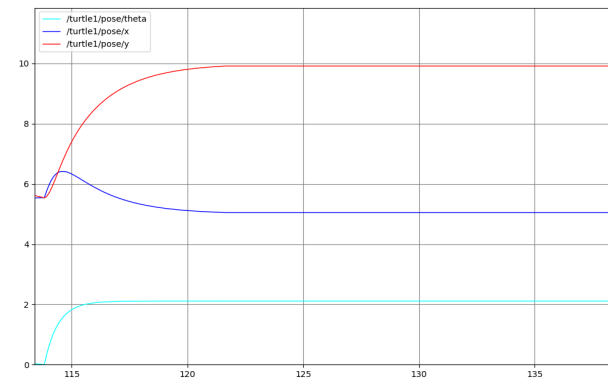


## Turtlesim Application : position control

---

- Turtle Sim Position
  - 실행 결과.

Target : [3,10]



END

# ROS tutorial 4

MORAI

## 목차

---

1. roslaunch
  - Roslaunch 기능
  - Launch 파일 작성 예제 1
  - Launch for turtlesim 1
  - Launch for turtlesim 2
2. Rosbag
  - Roslaunch 기능



## 1. roslaunch

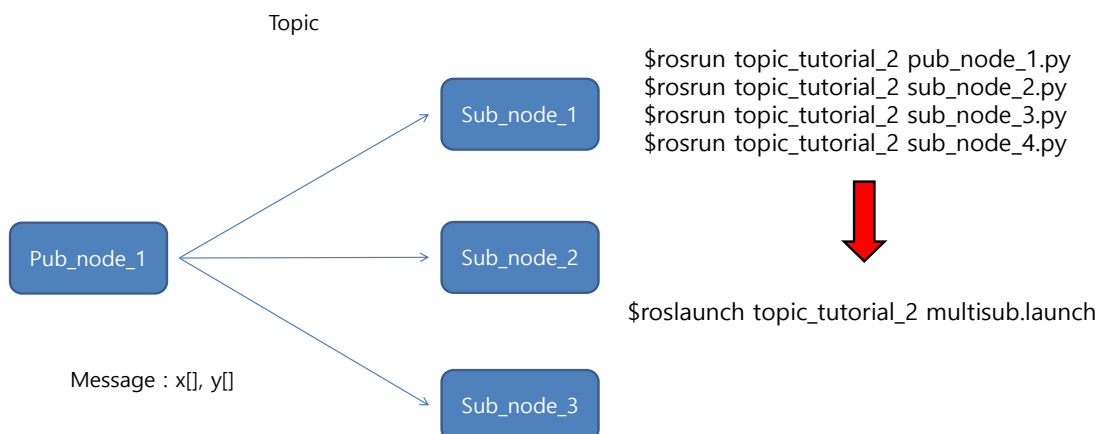
### roslaunch

---

- Roslaunch이란

- Rosrun : 하나의 노드를 실행하는 명령어
- Roslaunch : 하나 이상의 정해진 노드를 실행

\$ roslaunch [패키지 이름] [roslaunch 파일 이름]



## roslaunch

- Roslaunch이란

- <launch> : launch 파일을 시작. launch 파일은 반드시 <launch> 태그로 시작하여 </launch> 태그로 종료됩니다.
- <node> : 노드를 실행.
  - pkg="mypackage" : 노드가 포함되어 있는 패키지를 지정.
  - type="nodetype" : 노드 실행 파일 이름을 기입.
  - name="nodename" : 노드 이름을 지정.
- <machine> -> 노드를 실행시킬 기기 지정. 모든 노드가 로컬 네트워크에서 구동되는 경우 필요 없음.
- <include> -> launch 파일 내에서 다른 launch XML 파일 로드.
- <remap> -> 어떤 문자열을 전부 다른 문자열로 대응.
- <env> -> 이미 실행된 노드의 환경 변수를 설정.
- <rosparam> -> rosparam을 이용해서 ROS 파라미터를 설정.
- <group> -> 다수의 노드에 한꺼번에 설정을 적용시킬 때 사용.

[https://enssionaut.com/xe/board\\_robotics/974](https://enssionaut.com/xe/board_robotics/974)

## roslaunch

- Roslaunch 작성 예제

- topic\_tutorial\_2/launch/multisub.launch 작성.

```
1 <launch>
2
3 <!-- publisher node -->
4 <node pkg="topic_tutorial_2" type="pub_node_1.py" name="pub_node_1" />
5
6 <!-- subscriber node -->
7 <node pkg="topic_tutorial_2" type="sub_node_1.py" name="sub_node_1" />
8 <node pkg="topic_tutorial_2" type="sub_node_2.py" name="sub_node_2" />
9 <node pkg="topic_tutorial_2" type="sub_node_3.py" name="sub_node_3" />
10
11 <!-- visualisation with rqt -->
12 <node name="rqt_gui" pkg="rqt_gui" type="rqt_gui"/>
13
14 </launch>
15
```

## roslaunch

- Roslaunch 작성 예제
  - roslaunch topic\_tutorial\_2 multisub.launch
  - Loginfo가 보이지 않는다?

```
sjyoon@sjyoon-VirtualBox:~$ roslaunch topic_tutorial_2 multisub.launch
... logging to /home/sjyoon/.ros/log/40555040-95d0-11ea-8b86-08002743d974/roslaunch-sj
yoon-VirtualBox-3725.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://10.10.120.215:46873/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.3

NODES
/
  pub_node_1 (topic_tutorial_2/pub_node_1.py)
  rqt_gui (rqt_gui/rqt_gui)
  sub_node_1 (topic_tutorial_2/sub_node_1.py)
  sub_node_2 (topic_tutorial_2/sub_node_2.py)
  sub_node_3 (topic_tutorial_2/sub_node_3.py)

ROS_MASTER_URI=http://10.10.120.215:11311

process[pub_node_1-1]: started with pid [3744]
process[sub_node_1-2]: started with pid [3745]
process[sub_node_2-3]: started with pid [3746]
process[sub_node_3-4]: started with pid [3747]
process[rqt_gui-5]: started with pid [3748]
```

## roslaunch

- Roslaunch 작성 예제
  - 대안책1 : --screen
  - roslaunch topic\_tutorial\_2 multisub.launch --screen

```
sjyoon@sjyoon-VirtualBox:~$ roslaunch topic_tutorial_2 multisub.launch --screen
... logging to /home/sjyoon/.ros/log/40555040-95d0-11ea-8b86-08002743d974/roslaunch-sj
yoon-VirtualBox-6074.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://10.10.120.215:43795/

SUMMARY
=====

PARAMETERS
* /rostdistro: melodic
* /rosversion: 1.14.3

NODES
/
  pub_node_1 (topic_tutorial_2/pub_node_1.py)
  rqt_gui (rqt_gui/rqt_gui)
  sub_node_1 (topic_tutorial_2/sub_node_1.py)
  sub_node_2 (topic_tutorial_2/sub_node_2.py)
  sub_node_3 (topic_tutorial_2/sub_node_3.py)

ROS_MASTER_URI=http://10.10.120.215:11311

process[pub_node_1-1]: started with pid [6095]
process[sub_node_1-2]: started with pid [6096]
process[sub_node_2-3]: started with pid [6097]
process[sub_node_3-4]: started with pid [6098]
process[rqt_gui-5]: started with pid [6099]
[INFO] [1589453819.507418]: 15.0
[INFO] [1589453819.507452]: -1.0
[INFO] [1589453819.507735]: 9.0
[INFO] [1589453819.609028]: 15.0
[INFO] [1589453819.609072]: 9.0
[INFO] [1589453819.609847]: -1.0
```

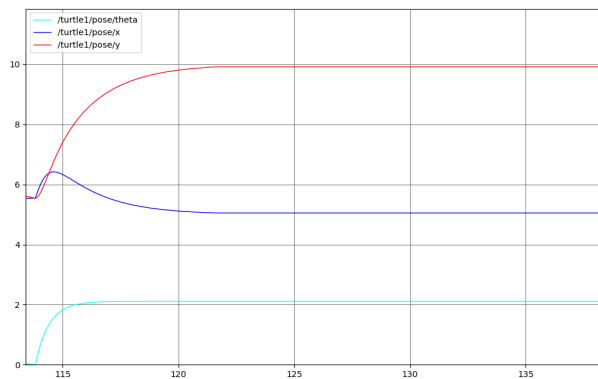
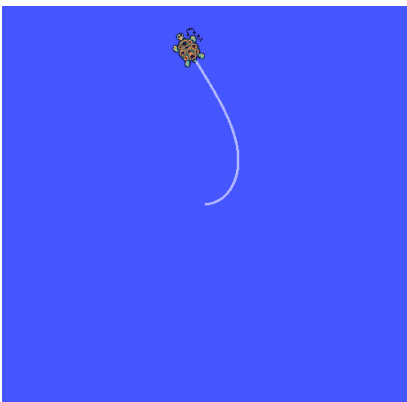
## roslaunch

- Roslaunch 작성 예제
  - 대안책2 : output="screen"

```
1 <launch>
2
3 <!-- publisher node -->
4 <node pkg="topic_tutorial_2" type="pub_node_1.py" name="pub_node_1" output="screen"/>
5
6 <!-- subscriber node-->
7 <node pkg="topic_tutorial_2" type="sub_node_1.py" name="sub_node_1" output="screen"/>
8 <node pkg="topic_tutorial_2" type="sub_node_2.py" name="sub_node_2" output="screen"/>
9 <node pkg="topic_tutorial_2" type="sub_node_3.py" name="sub_node_3" output="screen"/>
10
11 <!-- visualisation with rqt -->
12 <node name="rqt_gui" pkg="rqt_gui" type="rqt_gui"/>
13
14 </launch>
15
```

## roslaunch

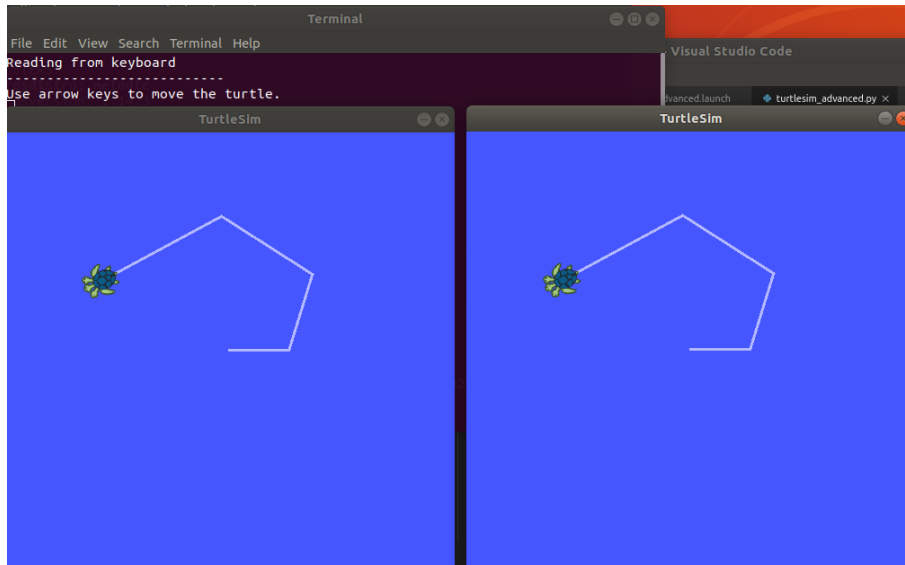
- Roslaunch turtlesim 응용 1
  - Turtlesim\_control 의 position control 노드와 rqt plot 노드를 동시에 실행시키는, launch 파일을 만드시오.



## roslaunch

---

- Roslaunch turtlesim 응용 2
  - Turtlesim\_node 두개와 teleop\_key, mimic 노드를 실행시키고, 하나의 turtle 이 다른 turtle 따라 하는 launch 파일 만들기.
  - Hint : <group> 사용



END

## ROS tutorial

프로젝트 지향 자율주행차 전문인력 양성과정

## ROS tutorial

프로젝트 지향 자율주행차 전문인력 양성과정